

REMARKS

Claims 1, 2, 4-9, 17-19, and 21-29 are all the claims presently pending in the application. Claims 3, 10-16, and 20 are canceled. Various claims have been amended to more particularly define the invention.

It is noted that Applicants specifically state that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

In the latest Office Action, the Examiner reopened prosecution and newly rejects claims 2, 5-9, 17-19, and 21-29 under 35 U.S.C. §112, second paragraph, for allegedly being indefinite, including indefiniteness due to various informalities. Applicants believe the above claim amendments appropriately address the Examiner's concerns and respectfully requests that the Examiner reconsider and withdraw these indefiniteness rejections.

Claims 17-19 stand rejected under 35 U.S.C. §102(e) as allegedly anticipated by US Patent 7,031,994 to Lao et al. Claims 1, 4, 23, 24, 26, 28, and 29 stand rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Lao, further in view of US Patent 5,025,407 to Gulley et al.

From the above-recited status, Applicants understand that claims 1, 4, 17-19, 23, 24, 26, 28, and 29 (and only these claims) stand rejected on prior art grounds. Therefore, Applicants understand that claims 2, 5-9, 21, 22, 25, and 27 are allowable if rewritten in independent format and the indefiniteness rejections based on informalities are cleared up.

These rejections are respectfully traversed in the following discussion.

I. THE CLAIMED INVENTION

The claimed invention, as described, for example, in independent claim 1, is directed to a computer including a processor, a memory system, a co-processing unit, and a plurality of data registers for data exchange with the co-processing unit. The computer is controlled to implement a method of increasing efficiency in executing a matrix operation that uses matrix data in a standard format, the standard format comprising one of a column major format and a row major format. The method comprises, for matrix data stored in the standard format in the memory system, wherein the matrix data comprises data of any of a complete matrix, a

complete submatrix, or a part of a matrix or submatrix, using the processor to separate the matrix data into blocks of data, each block having a size p-by-q.

The processor then rearranges and places in the memory system of the computer, for retrieval in a repetitive manner for executing the matrix operation, the blocks of data to be contiguous blocks of contiguous data such that the matrix data is represented in a nonstandard format that permits the matrix data to be moved from the memory system into a position for performing the matrix operation more quickly than if the matrix data had been moved as stored in the standard format.

In another aspect of the present invention described by independent claims 26 and 29, the present invention permits a hardware/instruction set deficiency to be overcome by using software instructions only. The software instructions would actually seem to be performing two errors relative to data for the intended operation, but the two errors together are designed to overcome the computer deficiency without having to redesign the machine and by using conventional compilers. In the exemplary embodiment, the deficiency involves the interface with the FPU, the co-processing unit that actually executes the matrix operation described, although the concept is clearly more general.

The present invention is actually only one of several ways used by the inventors to improve efficiency of matrix processing on the assignee's BlueGeneL computer. As explained beginning at line 19 on page 3, the present invention involves a method that improves efficiency and/or overcomes a hardware/instruction set deficiency without going through the expense of a re-design of computer hardware or instructions.

Conventional wisdom would have corrected the deficiency by redesigning the chip in the computer.

The claimed invention, on the other hand, provides a software solution that can be accomplished using conventional compilers and conventional assemblers. Applicants submit that this method is clearly non-obvious, since it involves intentionally executing two errors relative to the intended matrix operation on standard matrix data. That is, the present invention teaches a preliminary processing that converts the matrix data into non-standard format and a new processing added to the matrix operation so that the overall result will be correct for the matrix operation. In the example described in the disclosure, the second error involved loading data into the FPU registers using a non-standard loading instruction that crisscrosses the loading from normal word order.

However, it is noted that the feature described in independent claim 1 also has utility separate from overcoming a design deficiency, since the standard matrix format using row/column major format is typically a disadvantage for at least one of the three matrices involved in, for example, a matrix multiplication process.

Thus, the present invention provides a mechanism to overcome a hardware deficiency in a computer, using a software mechanism wherein data is re-arranged preliminarily and stored in memory in a format that is optimum to overcome the deficiency and increase processing efficiency, particularly when the re-arranged data is repetitively retrieved from memory for the processing. In the exemplary embodiment described in the disclosure, the data is matrix data to be used in linear algebra subroutines, the deficiency of the hardware is the interface at the FPU, and the matrix data is desired to be the transpose of submatrices only after having been finally loaded into the FPU.

However, it is noted that the matrix data stored in memory in the present invention is neither the original matrix data nor the transpose of the matrix data and is, therefore, certainly not stored in standard format.

The prior art of record has absolutely no suggestion whatsoever of this manner of storing matrix data, wherein matrix data is stored neither in original data nor in transpose format, nor even stored in standard format.

In general, the intermediate data format of the present invention is determined by considering what data format is optimal for the processing involved in the FPU and the nature of the hardware deficiency and is arrived at by working backwards from the processing at the point of the hardware deficiency.

II. THE PRIOR ART REJECTIONS

The Anticipation Rejection for Claims 17-19, Based on Lao et al (US Patent 7,031,994)

The Examiner considers that Lao et al anticipates the present invention described by claims 17-19, presumably because the exemplary embodiment in the present application demonstrates the method of the present invention as implemented to load the transpose of the matrix into the FPU.

However, Applicants respectfully submit that the method of the claimed invention is clearly patentably distinguishable from the transposition process described in Lao, particularly in view of the different problem being addressed by the present invention from that of Lao. More specifically, Lao addresses simply the process of matrix transposition as a solution to save memory space requirement (see lines 46-47 of column 1 of Lao).

To achieve this transposition, as clearly described even in the Abstract, Lao partitions the matrix data into blocks, moves these blocks row-wise into the cache, rewrites the data row-wise into a column of memory and uses a permutation vector to arrive at the transpose of the original matrix data and stores that transpose in memory.

Applicants submit that this simple matrix transposition fails to satisfy the plain meaning of the claim language of even the independent claims, particularly in view of the different purpose of the present invention, as articulated in the claims themselves. That is, in contrast to the simple transposition method of Lao, the exemplary embodiment described in the disclosure arrives at the transpose of portions of the matrix data only after it has been loaded into the FPU, and that transpose is achieved only after using the non-standard crisscross loading pattern (e.g., reference dependent claim 18).

Thus, in method of the present invention, data is prepared and stored in memory as predetermined to permit an optimal loading into FPU, based upon the hardware interface between the cache and FPU and is neither original matrix data nor its transpose. The independent claims of the present application actually refer to the “pseudo matrix” PA 605 shown in Figure 6 that is stored in memory to be repetitively retrieved for the matrix processing being executed in the FPU. Only upon retrieval and only after having been loaded into the FPU using a nonstandard crisscross loading pattern does this retrieved pseudo matrix data appear in the L0 registers of the FPU in the format of transposed matrix portions useful for the matrix processing described in the exemplary embodiment.

Therefore, the rearrangement of data in memory in the present invention does not result in the transpose of the matrix being stored in memory, as occurs in Lao. Rather, the pseudo matrix is stored in memory, to be retrieved repetitively for use by the FPU processing, as appropriate.

Lao makes no suggestion for storing matrix data in any format except the matrix transpose and, indeed, such matrix transposition is the entire purpose of Lao. Thus, Lao does not store anything in memory except either the original matrix data or the matrix transposed data, and both of these formats are stored in memory in standard format.

In contrast, the present invention stores a pseudo matrix in memory (for repetitive retrieval) and this data is not stored in standard format (see Figure 6). Because of this fundamental difference in purpose and the failure to store a pseudo matrix in nonstandard format, the technique of Lao fails to satisfy the plain meaning of the claim language, as follows.

Relative to independent claim 17, Lao stores in memory only either the original matrix data or the matrix transpose. There is nothing stored in memory in Lao that is in nonstandard format. In the environment of the L1 cache/FPU interface problem of the present invention, this simple transposed matrix data would have the same inefficiency noted by the present inventors, which was addressed using the software method of using two “errors” to place the matrix data into the FPU registers in the desired format of the transpose.

Therefore, assuming *arguendo* that the transposed matrix data of Lao were to be subsequently retrieved from memory for processing (as required by the plain meaning of the claim language), there is no suggestion in Lao to store blocks of data in nonstandard format (since transposed matrix data is still standard format). Nor is there any suggestion in Lao of an architecture for which matrix data stored in nonstandard format would be beneficial, as required by the final claim limitation.

It is further noted that, when Lao filed his patent, the hardware of current invention not yet present. That is, the problem being addressed by the present invention was not then present (e.g., Multiple loads out of cache into the FPU).

More specifically, the matrix processing of the present invention is being executed in the FPU. Even if the transposition processing in Lao were to be considered the “matrix operation” required by the language of claim 17, there is nothing in Lao corresponding to storing the matrix data in a memory in nonstandard format to be used for repetitive retrieval, since the processing of moving rows of data in Lao is occurring via the cache and the movement of these blocks of row data would not qualify to satisfy the plain meaning of the claim language requiring that data be stored in nonstandard format. The Examiner points to the movement of data demonstrated in column 16 of Lao that is used to explain the formation of a matrix transpose. However, this data movement is not equivalent to storing the data in nonstandard format, and the only significant storage of data in Lao is the storage of the transposed matrix, which is stored in standard format.

In summary, Lao is not intending to store matrix data in nonstandard format for purpose of performing a matrix operation, as required by the plain meaning of independent

claim 17. In contrast, the present invention actually intends to retrieve this pseudo matrix data repetitively as part of the linear algebra processing in the FPU, thereby adding to the efficiency of the method of the present invention.

The Examiner does not rely upon secondary reference Gulley for overcoming these deficiencies of Lao, so Gulley fails to overcome these deficiencies.

Hence, turning to the clear language of the claims, in Lao there is no teaching or suggestion of: "... to perform a method of storing information of a matrix in a register block data format: ... said register data block format representing the matrix data in a format that is no longer be in either of said standard column format or said standard row format", as required by independent claim 17.

Relative to the rejection for claims 18 and 19, wherein the Examiner relies upon the description in Lao at lines 37-58 of column 16, Applicants respectfully submits that this description relates to the transposition of matrix data in the abstract. There is no suggestion in these lines of loading data into FPU registers, let alone loading data in a non-standard crisscross pattern. The matrix transpose in Lao is loaded into memory itself, not data registers, and there is no crisscross loading pattern. Lao also does not address the problem of loading into FPU data registers and does not even mention loading into the CPU, which the site of its data manipulation, if the matrix transposition is considered to be a matrix processing in a broad construction.

The Obviousness Rejection for Claims 1, 4, 23, 24, 26, 28, and 29, Based on Lao,
Further in view of Gulley

The Examiner considers that Lao renders obvious claims 1, 4, 23, 24, 26, 28, and 29, even further modified in accordance with Gulley.

Applicants respectfully submit that the Examiner has failed to meet the initial burden of an obviousness rejection for these claims, since the entire purpose of Lao is to form the matrix transpose, using cache loading/unloading. There is no suggestion in Lao of the details of these claims.

The Examiner even concedes that Lao fails to describe a coprocessor. To overcome this deficiency, the Examiner relies upon newly-cited secondary reference Gulley. According to the Examiner, it would have been obvious to modify Lao in accordance with Gulley "... *in order to increase the speed of processing.*"

In response, Applicants respectfully point out that the modification required to satisfy the plain meaning of the claim language would be that of changing the transpose stored in memory in Lao into matrix data that is no longer stored in standard format. There is no suggestion to do so in either Lao or Gulley and no reason to do so by merely adding a coprocessor to increase the speed of processing.

Moreover, the increase in processing speed by adding a coprocessor to Lao would be for the execution of the linear algebra processing itself, not for a preliminary rearrangement of the matrix data stored in memory to be in a nonstandard format. The preliminary processing of Lao to form the transpose is executed by the CPU and there would be no additional speed involved in this processing even if a coprocessor were to be added to Lao.

In summary, Applicants submit that, although Lao and the present invention both describe matrix transposition, Lao simply achieves the matrix transpose in a manner that differs in its details of execution from that of the claimed invention. This different procedure is expected since Lao is not concerned with overcoming a problem of a hardware interface on its machine because at the time of Lao, even with a co-processor installed he could not have used the feature of crisscrossing of data loading, since such commands were not available. Moreover, even if this feature had been available and Lao had used it in subsequent processing using the transposed data, the data would have been in improper order in the FPU registers, so that this problem would have to be corrected in the registers of the FPU, using a register operation. In the BlueGeneL computer, such a register operation is available but it is too slow.

The present invention has resulted from this lack of a mechanism in the BlueGeneL to change the data back in the FPU registers quickly, without developing either new hardware or new FPU instructions.

Lao actually uses only a "slow" loading processing to get its data from memory into the CPU for its data transpose procedure and there would be no difference in retrieving data in this manner from memory into an FPU instead of the CPU. Therefore, Lao would have been faced with a time-consuming and slow correction of data within the FPU, and such correction processing would have been slower than simply using the slow loading that is occurring inherently in the processing described in Lao. Thus, Lao had no reason to make a preliminary conversion described by the present application. The mechanism of the present invention is actually a preliminary conversion of data that takes time to perform, but the savings in time occurs because data is repetitively retrieved in a fast retrieval based on having

reorganized the data in a nonstandard format that works in conjunction with features not available at the time of Lao.

Therefore, although the exemplary embodiment used the matrix transpose to describe its method, the transpose operation itself is not the key aspect of the method. The transpose in the present invention does not occur until the data is loaded into the FPU registers, using a non standard loading order of the data. The independent claims clearly address a novel data transformation that is stored in memory in nonstandard format.

Lao does not suggest using nonstandard format for data storage, let alone the nonstandard format described in the present invention as “pseudo matrix” data that is predetermined to provide a format of contiguous matrix data that will be subsequently loaded out- of- normal-word-order into the FPU, to result upon such nonstandard loading to be in the optimal format for the desired matrix processing. Coincidentally, the example described in the present application happened to demonstrate that the matrix transpose would result in the FPU registers once the nonstandard data is loaded in nonstandard order into the FPU registers.

Relative to independent claims 23 and 26 (and dependent claims 24, 25, and 27), there is no equivalent in Lao of a pseudo matrix data that is stored in nonstandard format and that is loaded into a processing unit, let alone a pseudo matrix predetermined as permitting an optimal loading and an optimal processing time for linear algebra processing. Lao also fails to describe a disadvantage with hardware or instructions, as required by claim 26, or of placing data into registers, as required by claims 24, 25, and 27.

Relative to independent claim 28 (and dependent claim 29), there are no errors being executed in Lao. Nor is there any hardware disadvantage being overcome relative to a matrix processing.

Relative to dependent claims 4 (and 8), the 2x2 block in Lao is used only for generating the transpose, which is stored in standard format. Because Lao fails to demonstrate each of the above-identified elements of the claims, the rejection currently of record fails to meet the initial burden of a *prima facie* rejection.

Therefore, Applicant submits that there are elements of the claimed invention that are not taught or suggest by Lao, and the Examiner is respectfully requested to withdraw this rejection.

III. FORMAL MATTERS AND CONCLUSION

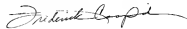
In view of the foregoing, Applicants submit that claims 1, 2, 4-9, 17-19, and 21-29, all the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance.

The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,



Date: February 28, 2008

Frederick E. Cooperrider
Registration No. 36,769

McGinn Intellectual Property Law Group, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

CERTIFICATION OF TRANSMISSION

I certify that I transmitted electronically, via EFS, this Amendment under 37 CFR §1.111 to the USPTO on February 28, 2008.



Frederick E. Cooperrider
Reg. No. 36,769